

Min-plus System Theory Applied to Communication Networks

Jean-Yves Le Boudec and Patrick Thiran

LCA - ISC

School of Computer and Communication Sciences

EPFL

CH-1015 Lausanne

Switzerland

Abstract

Network Calculus is a set of recent developments, which provide a deep insight into flow problems encountered in networking. It can be viewed as the system theory that applies to computer networks. Contrary to traditional system theory, it relies on max-plus and min-plus algebra. In this paper, we show how a simple but important fixed-point theorem (residuation theorem in min-plus algebra) can be successfully applied to a number of problems in networking, such as window flow control, multimedia smoothing, and bounds on loss rates.

1 Introduction

Network calculus is a theory of (mainly deterministic) queuing systems found in computer networks. The foundation of network calculus lies in the mathematical theory of dioids, and in particular, the Min-Plus dioid (also called Min-Plus algebra). With network calculus, we are able to understand some fundamental properties of integrated services networks, of window flow control, of scheduling and of buffer or delay dimensioning.

The main difference between network calculus, which can be regarded as the system theory that applies to computer networks, and traditional system theory, as the one which was so successfully applied to design electronic circuits, is the algebra under which operations take place: addition becomes now computation of the minimum, multiplication becomes addition.

Let us illustrate this difference with an example. Consider a very simple circuit, such as the RC cell represented in Figure 1. If the input signal is the voltage $x(t) \in \mathbb{R}$, then the output $y(t) \in \mathbb{R}$ of this simple circuit is the convolution of x by the impulse response of this circuit, which is here $h(t) = \exp(-t/RC)/RC$ for $t \geq 0$:

$$y(t) = (h \otimes x)(t) = \int_0^t h(t-s)x(s)ds.$$

Consider now a node of a communication network, which is idealized as a (greedy) shaper. A (greedy) shaper is a device that forces an input flow $x(t)$ to have an output $y(t)$ that conforms to a given set of rates according to a traffic envelope σ (the shaping curve), at the expense of possibly delaying bits in the buffer. Here the input and output ‘signals’ are

cumulative flows, defined as the number of bits seen on the data flow in time interval $[0, t]$. These functions are non-decreasing with time t . We will denote by \mathcal{G} the set of non-negative wide-sense increasing functions and by \mathcal{F} denote the set of wide-sense increasing functions (or sequences) such that $f(t) = 0$ for $t < 0$. Parameter t can be continuous or discrete. We will see in this paper that x and y are linked by the relation

$$y(t) = (\sigma \otimes x)(t) = \inf_{s: 0 \leq s \leq t} \{\sigma(t - s) + x(s)\}. \quad (1.1)$$

This relation defines the min-plus convolution between σ and x .

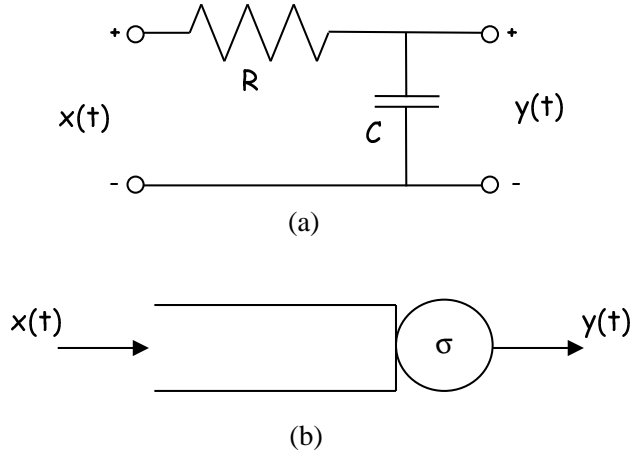


Figure 1: Traditional system theory for an elementary circuit (top) and min-plus system theory for a shaper (bottom).

We first review the basic concepts of network calculus, namely the way we characterize the ‘signals’ (i.e. the flows) via arrival curves (Section 2) and the ‘system’ (e.g., the network node) via a service curve (Section 3). These tools will enable us to derive some deterministic performance bounds on quantities such delays and backlogs (Section 4), which are defined as follows, for a lossless system with input flow $x(t)$ and output flow $y(t)$:

Definition 1.1 (Backlog and Delay). *The backlog at time t is $x(t) - y(t)$, the virtual delay at time t is*

$$d(t) = \inf \{\tau \geq 0 : x(t) \leq y(t + \tau)\}.$$

The backlog is the amount of bits that are held inside the system; if the system is a single buffer, it is the queue length. In contrast, if the system is more complex, then the backlog is the number of bits “in transit”, assuming that we can observe input and output simultaneously. The virtual delay at time t is the delay that would be experienced by a bit arriving at time t if all bits received before it are served before it. If we plot $x(t)$ and $y(t)$

versus t , the backlog is the vertical deviation between these two curves. The virtual delay is the horizontal deviation.

The second part of the paper (Sections 5 and 6) is a systematic method for modeling a number of situations arising in communication networks (lossless greedy shaping, window flow control, video smoothing, upper bound on loss rates, packetizing variable length packets), as sets of inequalities using min-plus operators. We will find the maximal solution of these systems of inequalities using a central result of min-plus algebra which describes the maximal solution of these systems of inequalities using the concept of closure of an operator [3] (Theorem 7.1 in Section 7). It is equivalent to the description of a system by a set of ordinary differential equations in traditional system theory.

The interested reader is also referred to the pioneering work of Cruz [8], Chang[5], Agrawal and Rajan[1], as well as to [6, 10].

2 Arrival curves

To provide guarantees to data flows requires some specific support in the network; as a counterpart, the traffic sent by sources needs to be limited. This is done by using the concept of arrival curve, defined below.

Definition 2.1 (Arrival Curve). *Given a wide-sense increasing function α defined for $t \geq 0$ (namely $\alpha \in \mathcal{F}$), we say that a flow x is constrained by α if and only if for all $s \leq t$:*

$$x(t) - x(s) \leq \alpha(t - s).$$

Note that this is equivalent to imposing that for all $t \geq 0$

$$x(t) \leq \inf_{0 \leq s \leq t} \{\alpha(t - s) + x(s)\} = (\alpha \otimes x)(t) \quad (2.2)$$

The simplest arrival curve is $\alpha(t) = Rt$. Then the constraint means that, on any time window of width τ , the number of bits for the flow is limited by $R\tau$. We say in that case that the flow is peak rate limited. This occurs if we know that the flow is arriving on a link whose physical bit rate is limited by R bits/sec. A flow where the only constraint is a limit on the peak rate is often (improperly) called a “constant bit rate” (CBR) flow.

More generally, because of their relationship with leaky buckets, we will often use *affine* arrival curves $\gamma_{r,b}$, defined by: $\gamma_{r,b}(t) = rt + b$ for $t > 0$ and 0 otherwise. Having $\gamma_{r,b}$ as an arrival curve allows a source to send b bits at once, but not more than r bits/s over the long run. Parameters b and r are called the burst tolerance (in units of data) and the rate (in units of data per time unit). The Integrated services framework of the Internet (Intserv) uses arrival curves, such as

$$\alpha(t) = \min\{M + pt, rt + b\} = \gamma_{p,M}(t) \wedge \gamma_{r,b}(t)$$

where M is interpreted as the maximum packet size, p as the peak rate, b as the burst tolerance, and r as the sustainable rate Figure 2. Notation \wedge stands for minimum or infimum.

In Intserv jargon, the 4-uple (p, M, r, b) is also called a T-SPEC (traffic specification). ATM uses similar curves.

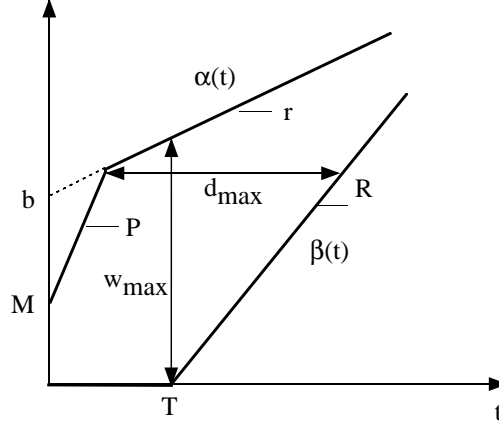


Figure 2: Arrival curve α for ATM VBR and for Intserv flows, rate-latency service curve β and vertical and horizontal deviations between both curves.

One can always replace an arrival curve α by its sub-additive closure, which is defined as

$$\bar{\alpha} = \inf\{\delta_0, \alpha, \alpha \otimes \alpha, \dots, \alpha^{(n)}, \dots\}$$

where $\alpha^{(n)} = \alpha \otimes \dots \otimes \alpha$ (n times) and δ_0 is the “impulse” function defined by $\delta_0(t) = \infty$ for $t > 0$ and $\delta_0(0) = 0$. One can show indeed that $x \leq x \otimes \alpha$ if and only if $x \leq x \otimes \bar{\alpha}$. If $\alpha(0) = 0$ and α is sub-additive (meaning that for all $s, t \geq 0$, $\alpha(s+t) \leq \alpha(s) + \alpha(t)$), then $\bar{\alpha} = \alpha$. As an example, one can check that $\bar{\gamma}_{r,b} = \gamma_{r,b}$.

Finally, it is possible to compute from measurements of a given flow $x(t)$ its minimal arrival curve, which is $(x \otimes x)(t)$ where \otimes denotes the min-plus deconvolution operator defined by

$$(x \otimes \sigma)(t) = \sup_{u \geq 0} \{x(t+u) - \sigma(u)\}, \quad (2.3)$$

for a given function $\sigma \in \mathcal{F}$. Note that if $x, \sigma \in \mathcal{F}$, then $(x \otimes \sigma) \in \mathcal{F}$ but in general $(x \otimes \sigma) \notin \mathcal{F}$ (it belongs to \mathcal{G}). One can check however that $(x \otimes x) \in \mathcal{F}$. Let us also mention that the name deconvolution is justified by the fact that for any $x, y, z \in \mathcal{F}$, $x \leq y \otimes z$ if and only if $x \otimes z \leq y$.

3 Service curves

We have seen that one first principle in integrated services networks is to put arrival curve constraints on flows. In order to provide reservations, network nodes in return need to offer some guarantees to flows. This is done by packet schedulers. The details of packet scheduling are abstracted using the concept of service curve, which we introduce in this section.

Definition 3.1 (Service Curve). Consider a system \mathcal{S} and a flow through \mathcal{S} with input and output function x and y . We say that \mathcal{S} offers to the flow a service curve β if and only if for all $t \geq 0$, there exists some $t_0 \geq 0$, with $t_0 \leq t$, such that

$$y(t) - x(t_0) \geq \beta(t - t_0).$$

Again, we can recast this definition as

$$y(t) \geq \inf_{0 \leq s \leq t} \{\beta(t - s) + x(s)\} = (\beta \otimes x)(t) \quad (3.4)$$

Let us consider a few examples. A simple one is a GPS (Generalized Processor Sharing) node which, by offering a service curve $\beta(t) = Rt$, guarantees that each flow is served at least at rate R bits/s during a busy period.

A second example is a guaranteed delay node. Here the only information we have about the network node is that the maximum delay for the bits of a given flow x is bounded by some fixed value T , and that the bits of the flow are served in first in, first out order. This is used with a family of schedulers called “earliest deadline first” (EDF), and can be translated as $y(t) \geq x(t - T)$ for all $t \geq T$. Using the “impulse” function δ_T defined by $\delta_T(t) = 0$ if $0 \leq t \leq T$ and $\delta_T(t) = +\infty$ if $t > T$, we have that $(x \otimes \delta_T)(t) = x(t - T)$. We have therefore shown that a guaranteed delay node offers a service curve $\beta = \delta_T$.

As a last example, the IETF assumes that RSVP routers offer a service curve of the form

$$\beta_{R,T}(t) = R[t - T]^+ = \begin{cases} R(t - T) & \text{if } t > T \\ 0 & \text{otherwise} \end{cases}$$

as shown on Figure 2. We call this curve the rate-latency service curve.

Finally, let us mention the following result, which is well-known in traditional system theory, and which is easy to establish in network calculus:

Theorem 3.1 (Concatenation of Nodes). Assume a flow traverses systems \mathcal{S}_1 and \mathcal{S}_2 in sequence. Assume that \mathcal{S}_i offers a service curve of β_i , $i = 1, 2$ to the flow. Then the concatenation of the two systems offers a service curve of $\beta_1 \otimes \beta_2$ to the flow.

As an example, consider two nodes offering each a rate-latency service curve β_{R_i, T_i} , $i = 1, 2$, as is commonly assumed with Intserv. A simple computation gives

$$\beta_{R_1, T_1} \otimes \beta_{R_2, T_2} = \beta_{R_1 \wedge R_2, T_1 + T_2}. \quad (3.5)$$

Thus concatenating RSVP routers amounts to adding the latency components and taking the minimum of the rates.

We are now also able to give another interpretation of the rate-latency service curve model. We can compute that $\beta_{R,T} = \delta_T \otimes \gamma_{R,0}$; thus we can view a node offering a rate-latency service curve as the concatenation of a guaranteed delay node, with delay T and a CBR or GPS node with rate R .

4 Three basic bounds

In this section we see the main simple network calculus results. They are all bounds for lossless systems with service guarantees [1]. The proofs are straightforward applications of the definitions of service and arrival curves.

The first theorem says that the backlog is bounded by the vertical deviation between the arrival and service curves:

Theorem 4.1 (Backlog Bound). *Assume a flow, constrained by arrival curve α , traverses a system that offers a service curve β . The backlog $x(t) - y(t)$ for all t satisfies:*

$$x(t) - y(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\}$$

We now use the concept of horizontal deviation. Call $\Delta(s) = \inf \{\tau \geq 0 : \alpha(s) \leq \beta(s + \tau)\}$. From Definition 1.1, $\Delta(s)$ is the virtual delay for a hypothetical system which would have α as input and β as output, assuming that such a system exists. Let $h(\alpha, \beta)$ be the supremum of all values of $\Delta(s)$. The second theorem gives a bound on delay for the general case.

Theorem 4.2 (Delay Bound). *Assume a flow, constrained by arrival curve α , traverses a system that offers a service curve of β . The virtual delay $d(t)$ for all t satisfies: $d(t) \leq h(\alpha, \beta)$.*

Theorem 4.3 (Output Flow). *Assume a flow, constrained by arrival curve α , traverses a system that offers a service curve of β . The output flow is constrained by the arrival curve $\alpha^* = \alpha \oslash \beta$.*

As a first application of the previous results, consider a VBR flow, defined by TSPEC (M, p, r, b) (hence $\alpha(t) = \{M + pt\} \wedge \{rt + b\}$) and served in one node which guarantees a service curve equal to the rate-latency function $\beta(t) = R[t - T]^+$. This example is the standard model used in Intserv (Figure 2). Let us apply Theorems 4.1 and 4.2. Assume that $R \geq r$ namely the reserved rate is as large as the sustainable rate of the flow. The buffer required for the flow is bounded by

$$w_{\max} = b + r \max \left(\frac{b - M}{p - r}, T \right)$$

The maximum delay for the flow is bounded by

$$d_{\max} = \frac{M + \frac{b-M}{p-r}(p-R)^+}{R} + T.$$

As a second application, let us show how these bounds, combined with Theorem 3.1, allow us to understand a phenomenon known in the Intserv community as “Pay Bursts Only Once”. Consider the concatenation of two nodes offering each a rate-latency service curve

β_{R_i, T_i} , $i = 1, 2$, as is commonly assumed with Intserv. Assume the fresh input is constrained by $\gamma_{r,b}$. Assume that $r < R_1$ and $r < R_2$. We are interested in the delay bound, which we know is a worst case. Let us compare the results obtained by applying Theorem 4.2 (i) to the network service curve (3.5), resulting in a delay bound D_0 ; (ii) iteratively to every node, resulting in two individual bounds D_1 and D_2 .

(i) The delay bound D_0 can be computed by application of Theorem 4.2:

$$D_0 = \frac{b}{R_1 \wedge R_2} + T_1 + T_2.$$

(ii) Now apply the second method. A bound on the delay at node 1 is (Theorem 4.2): $D_1 = b/R_1 + T_1$. The output of the first node is constrained by $\alpha^*(t) = b + rt + rT_1$, because of Theorem 4.3. A bound on the delay at the second buffer is therefore $D_2 = (b + rT_1)/R_2 + T_2$. Consequently,

$$D_1 + D_2 = \frac{b}{R_1} + \frac{b + rT_1}{R_2} + T_1 + T_2$$

It is easy to see that $D_0 < D_1 + D_2$. In other words, the bounds obtained by considering the global service curve are better than the bounds obtained by considering every buffer in isolation.

5 Some deterministic flow problems in networking

We now describe a few examples of situations arising in the networking context, which we will model as min-plus systems in the next section.

Example 1: Greedy lossless traffic shaper

We call *policer* with curve σ a device that counts the bits arriving on an input flow and decides which bits conform with an arrival curve of σ . We call *shaper*, with shaping curve σ , a bit processing device that forces its output to have σ as arrival curve. We call *greedy shaper* a shaper which delays the input bits in a buffer, whenever sending a bit would violate the constraint σ , but outputs them as soon as possible.

With ATM and sometimes with IP (Diffserv and Intserv), traffic sent over one connection, or flow, is policed at the network boundary. Policing is performed in order to guarantee that users do not send more than specified by the contract of the connection. Policing devices inside the network are normally buffered, they are thus shapers. Shaping is also often needed because the output of a buffer normally does not conform any more with the traffic contract specified at the input.

If $a(t)$ denotes the input flow, and if σ denotes the shaping curve (since it is an arrival curve, we can always assume that σ is sub-additive with $\sigma(0) = 0$, as explained in Section 2),

the output $x(t)$ of the shaper is therefore such that

$$\begin{aligned} x(t) &\leq a(t) \\ x(t) - x(s) &\leq \sigma(t - s) \end{aligned}$$

for all $0 \leq s \leq t$. The shaper is greedy if x is the maximal solution of this set of inequalities.

Example 2: Window flow controller

This example is found independently in [4] and [2]. A data flow $a(t)$ is fed via a window flow controller to a network offering a service curve of β . The window flow controller limits the amount of data admitted into the network in such a way that the total backlog is less than or equal to W , where W (the window size) is a fixed number (Figure 3).

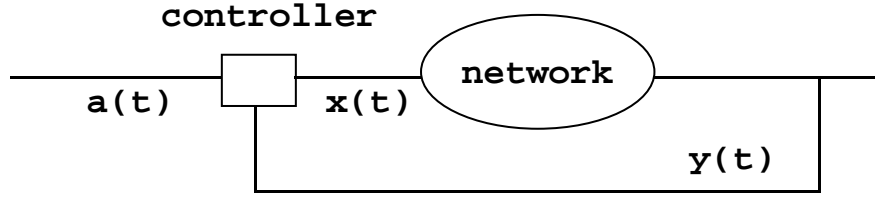


Figure 3: Example 2, from [4] or [2]

Call $x(t)$ the flow admitted to the network, and $y(t)$ the output. The definition of the controller means that $x(t)$ is the maximum solution to

$$\begin{cases} x(t) \leq a(t) \\ x(t) \leq y(t) + W \end{cases} \quad (5.6)$$

which implies that $x(t) = a(t) \wedge (y(t) + W)$. Note that we do not know the mapping $x(t) \rightarrow y(t)$, but we do know that

$$y(t) \geq (\beta \otimes x)(t). \quad (5.7)$$

In [4], (5.6) and (5.7) are used to derive that

$$x \geq \overline{(\beta + W)} \otimes a \quad (5.8)$$

Equation (5.8) means that the complete system offers a service curve equal to the sub-additive closure $\overline{(\beta + W)}$. We will show that this result can be obtained by min-plus methods.

Example 3: Optimal smoother

We consider the transmission of *variable bit rate* (VBR) video over a network offering a guaranteed service such as ATM VBR or the guaranteed service of the IETF (see Figure 4).

The guaranteed service requires that the flow produced by the output device conform with an arrival curve σ ; in return, it receives a service guarantee expressed by a network service curve β . Functions σ and β are derived from the parameters used for setting up the reservation, for example, from the T-SPEC and R-SPEC fields used with the Resource Reservation Protocol (RSVP). In order to satisfy the smoothness (arrival curve) constraint, the output of the encoder a is fed to a smoother, possibly with some look-ahead. The resulting stream $x(t)$ is transported by the network; at the destination, the decoder waits for an initial *playback delay* D and reads the stream $a(t - D)$ from the receive buffer.

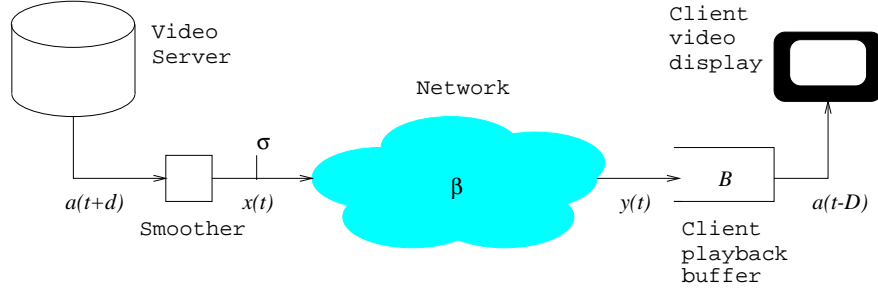


Figure 4: Example 3, optimal smoothing. In the figure, d denotes the amount of look-ahead allowed (here we consider that d is not limited and can be as large as the duration of the entire video trace).

We would like to minimize the playback delay D and the buffer size B at the receiver. Contrary to the shaper of Example 1, we allow our smoothing strategy to look-ahead and send some frames ahead of schedule, which a shaper does not.

Taking the convention that the time instant at which transmission begins is $t = 0$ (so that $x(t) = 0$ for $t \leq 0$), the constraints on the smoothed flow x fed in the network are therefore

$$x(t) \leq \delta_0(t) \quad (5.9)$$

$$x(t) \leq (x \otimes \sigma)(t) \quad (5.10)$$

whereas the ones on the flow y at the output of the network are

$$y(t) \geq a(t - D) \quad (5.11)$$

$$y(t) \leq a(t - D) + B. \quad (5.12)$$

Again, we do not know the exact mapping $x \rightarrow y$, but we know that the network offers a service curve β , so that $(x \otimes \beta)(t) \leq y(t) \leq x(t)$. Using the deconvolution operator \oslash and its properties, the set of constraints (5.11) and (5.12) can therefore be replaced by

$$x(t) \geq (a \oslash \beta)(t - D) \quad (5.13)$$

$$x(t) \leq a(t - D) + B. \quad (5.14)$$

We will compute the smallest values of D and B that guarantee a lossless smoothing, and propose one smoothing strategy (which is not unique: others are proposed in [11, 12], achieving the same optimal values of D and B).

Example 4: Losses in a shaper with finite buffer

We reconsider Example 1, but now we suppose that the buffer is not large enough to avoid losses for all possible input traffic, and we would like to compute the amount of data lost at time t , with the convention that the system is empty at time $t = 0$. We model losses as shown in Figure 5, where $x(t)$ is the data that has actually entered the system in the time interval $[0, t]$. The amount of data lost during the same period is therefore $L(t) = a(t) - x(t)$. The amount of data $(x(t) - x(s))$ that actually entered the system in any time interval $(s, t]$ is always bounded above by the total amount of data $(a(t) - a(s))$ that has arrived in the system during the same period. Therefore, for any $0 \leq s \leq t$, $x(t) \leq x(s) + a(t) - a(s)$ or equivalently

$$x(t) \leq \inf_{u \text{ such that } 0 \leq u \leq t} \{x(u) + a(t) - a(u)\}. \quad (5.15)$$

On the other hand, x is the part of a that does actually enter the shaper, so the output of the shaper is $y = \sigma \otimes x$. There is no loss for $x(t)$ if $x(t) - y(t) \leq X$ for any t . Thus

$$x(t) \leq y(t) + X = (\sigma \otimes x)(t) + X \quad (5.16)$$

The data x that actually enters the system is therefore the maximum solution to (5.15) and (5.16). In this paper we will provide an exact representation of $L(t)$, which we will use to obtain an upper bound \hat{l} on the loss ratio $L(t)/a(t)$.

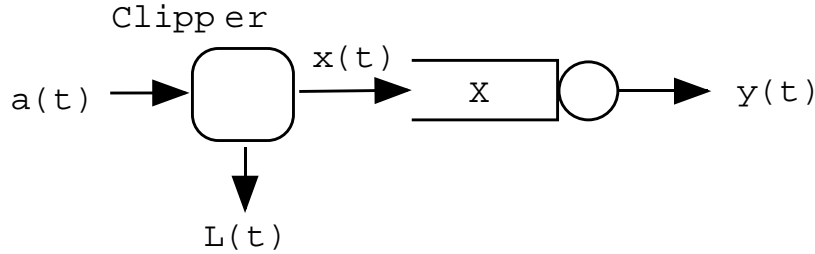


Figure 5: Example 4, shaper with losses

Example 5: Packetized greedy shapers

In real networks, variables representing cumulative flows are not continuous functions of time, because there is a minimum granularity (bit, cell, packet) that needs to be taken into account. Moreover, in some networks such as IP networks, packets have a variable length, which adds some additional subtleties [5, 9]. We need first to introduce a sequence L of cumulative packet lengths, which is a wide-sense increasing sequence with $L(0) = 0$ such that $l_{\max} = \sup_{n \in \mathbb{N}} \{L(n+1) - L(n)\}$ is finite. The length of the n -packet is $l_n = L(n) - L(n-1)$. A *L-packetizer* transforms a fluid input $R(t)$ into a *L-packetized* flow $\mathcal{P}_L(R)(t)$, where $\mathcal{P}_L(x) = \inf_{n \in \mathbb{N}} \{L(n) 1_{\{L(n) > x\}}\}$. In this expression, 1_A denotes the indicator

function of event A , namely here $1_{\{L(n) > x\}} = 1$ if $L(n) > x$ and 0 otherwise. We say that a flow $R(t)$ is L -packetized if $\mathcal{P}_L(R)(t) = R(t)$ for all t . For example, the output $x(t)$ of the packetizer in Figure 6 is L -packetized, but not its input $x'(t)$.

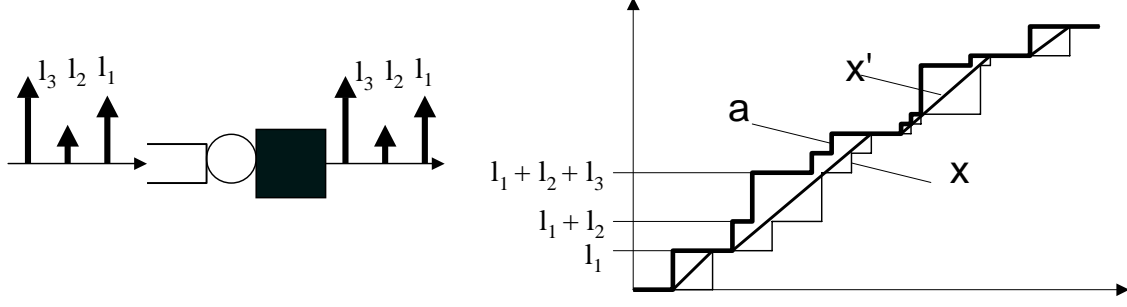


Figure 6: Example 5, concatenation of a constant bit rate greedy shaper and a packetizer (left). The input is $a(t)$, the output of the greedy shaper is $x'(t)$ is fed as input of the packetizer, whose (final) output is $x(t)$.

A *packetized shaper*, with shaping curve σ , forces its output to have σ as arrival curve and to be L -packetized. A *packetized greedy shaper* delays the input packets in a buffer, whenever sending the packet would violate the constraint σ , but outputs them as soon as possible. If $a(t)$ is the input flow, the output $x(t)$ is therefore the maximal solution of

$$\begin{aligned} x(t) &\leq a(t) \\ x(t) - x(s) &\leq \sigma(t - s) \\ x(t) &= \mathcal{P}_L(x)(t) \end{aligned}$$

for all $0 \leq s \leq t$.

6 System Modelling

The examples above involve particular types of operators $\Pi : \mathcal{F} \rightarrow \mathcal{G}$, which are

- Min-plus convolution: $\mathcal{C}_\sigma(x)(t) = (\sigma \otimes x)(t) = \inf_{0 \leq s \leq t} \{\sigma(t - s) + x(s)\}$,
- Min-plus deconvolution: $\mathcal{D}_\sigma(x)(t) = (x \oslash \sigma)(t) = \sup_{u \geq 0} \{x(t + u) - \sigma(u)\}$,
- Linear idempotent operator: $h_\alpha(x)(t) = \inf_{0 \leq s \leq t} \{\alpha(t) - \alpha(s) + x(s)\}$,
- Packetization: $\mathcal{P}_L(x)(t) = \inf_{n \in \mathbb{N}} \{L(n)1_{L(n) > x(t)}\}$.

We also define a set of properties, which are direct applications of [3]:

- Π is *isotone* if $x(t) \leq y(t)$ for all t always implies $\Pi(x)(t) \leq \Pi(y)(t)$ for all t . One can check that all four operators $\mathcal{C}_\sigma, \mathcal{D}_\sigma, h_\alpha, \mathcal{P}_L$ are isotone.
- Π is *causal* if for all t , $\Pi(x)(t)$ depends only on $x(s)$ for $0 \leq s \leq t$. $\mathcal{C}_\sigma, h_\alpha, \mathcal{P}_L$ are causal, but not \mathcal{D}_σ .
- Π is *upper-semi-continuous* if for any decreasing sequence of trajectories $(x^i(t))$ we have $\inf_i \Pi(x^i) = \Pi(\inf_i x^i)$. $\mathcal{C}_\sigma, h_\alpha, \mathcal{P}_L$ are upper semi-continuous, but not \mathcal{D}_σ .
- Π is *shift-invariant* if $y(t) = \Pi(x)(t)$ for all t and $x'(t) = x(t+s)$ for some s implies that for all t $\Pi(x')(t) = y(t+s)$. \mathcal{C}_σ and \mathcal{D}_σ are shift-invariant, but not h_α nor \mathcal{P}_L .
- Π is *min-plus linear* if it is upper-semi-continuous and $\Pi(x+K) = \Pi(x) + K$ for all constant K . $\mathcal{C}_\sigma, h_\alpha$ are min-plus linear, but not \mathcal{D}_σ nor \mathcal{P}_L .

We recast now the five examples using these operators.

Example 1: Greedy shaper. Its output is the maximum solution to

$$x \leq a \wedge \mathcal{C}_\sigma(x). \quad (6.17)$$

Example 2: Window flow controller. Define Π as the operator that maps $x(t)$ to $y(t)$. From Equation (5.6), we derive that $x(t)$ is the maximum solution to

$$x \leq a \wedge (\Pi(x) + W) \quad (6.18)$$

The operator Π can be assumed to be isotone, causal and upper-semi-continuous, but not necessarily linear. However, we know that $\Pi \geq \mathcal{C}_\beta$. We will exploit this formulation in Section 7.

Example 3: Optimal smoother. Constraints (5.9) to (5.14) can be recast as

$$x(t) \leq \delta_0(t) \wedge \mathcal{C}_\sigma(x)(t) \wedge \{a(t-D) + B\} \quad (6.19)$$

$$x(t) \geq \mathcal{D}_\beta(a)(t-D) \quad (6.20)$$

Example 4: Losses in a shaper with finite buffer. All operators are linear. We know that $x \leq a$. Combining this relation with (5.15) and (5.16), we derive that x is the maximum solution to

$$x \leq a \wedge h_a(x) \wedge (\sigma \otimes x + X). \quad (6.21)$$

Example 5: Greedy packetized shaper. Its output of the function $x \in \mathcal{F}$ which (i) is L-packetized and (ii) is the maximal solution of

$$x \leq a \wedge \mathcal{C}_\sigma(x) \wedge \mathcal{P}_L(x). \quad (6.22)$$

7 Space Method

In this paper we apply Theorem 4.70, item 6 [3] to the problems formulated in the previous section.

Theorem 7.1. *Let Π be an operator $\mathcal{F} \rightarrow \mathcal{G}$, and assume it is isotone and upper-semi-continuous. For any fixed function $a \in \mathcal{F}$, the problem*

$$x \leq \Pi(x) \wedge a \tag{7.23}$$

has one maximum solution, given by

$$x = \overline{\Pi}(a) = \inf \{a, \Pi(a), \Pi[\Pi(a)], \dots, \Pi^{(n)}(a), \dots\}.$$

The theorem is proven in [3], though with some amount of notation, using the fixed point method. It can also easily be proven directly as follows. Consider the sequence of decreasing sequences defined by $x^0 = a$ and $x^{n+1} = \Pi(x^n) \wedge x^n$, $n \in \mathbb{N}$. Then $x^* = \inf_{n \in \mathbb{N}} x^n$ is a solution of (7.23) because Π is upper-semi-continuous. One easily check that $x_n = \inf_{0 \leq m \leq n} \{\Pi^m(a)\}$, so that $x^* = \overline{\Pi}(a)$. Conversely, if x is a solution, then $x \leq x^n$ for all n because Π is isotone and thus $x \leq x^*$. We call the application of this theorem the space method, because it is based on an iteration on complete trajectories $x(t)$.

Let us first apply the theorem to **Example 1**. The maximal solution of (6.17) is

$$x = \overline{\mathcal{C}}_\sigma(a) = \overline{(\sigma \otimes a)} = \overline{\sigma} \otimes a = \sigma \otimes a,$$

the latter equality resulting from the sub-additivity of σ . An immediate consequence of this result is that a greedy shaper offers to the incoming flow a service curve equal to σ . The input-output characterization of greedy shapers $x = \sigma \otimes a$ is however much stronger than the service curve property, since we have here an equality instead of a lower bound.

Let us next apply the theorem to **Example 2**. We know now that (6.18) has one maximum solution and that it is given by

$$x = \overline{(\Pi + W)}(a).$$

Now from (5.7) we have $\Pi(x) + W \geq \beta \otimes x + K$. One easily shows that $x \geq \overline{(\beta + W)} \otimes a$ which is Equation (5.8).

The maximal solution of (6.19) and (6.20) in **Example 3**, which is

$$x(t) = \overline{\mathcal{C}}_\sigma(\delta_0(t) \wedge \{a(t - D) + B\}) = \sigma(t) \wedge \{(\sigma \otimes a)(t - D) + B\} \tag{7.24}$$

is guaranteed to exist if and only if it always verifies (6.20), which yields that

$$\begin{aligned} \sup_t \{(a \otimes \beta)(t - D) - \sigma(t)\} &\leq 0 \\ \sup_t \{(a \otimes \beta)(t - D) - (\sigma \otimes a)(t - D)\} &\leq B. \end{aligned}$$

Working out these inequalities, we find that the smallest playback delay and buffer size are

$$\begin{aligned} D &= \inf \{s \geq 0 : (a \odot (\beta \otimes \sigma))(-s) \leq 0\} \\ B &= ((a \odot a) \odot (\beta \otimes \sigma))(0). \end{aligned}$$

These optimal values are achieved by a smoothing strategy implemented at the sender side, given by (7.24).

From Theorem 7.1, the maximal solution of (6.21) in **Example 4** is $x = \overline{h_a \wedge \mathcal{C}_{\sigma_X}}(a)$, which after some manipulations (see [10]) can be recast as

$$x = \overline{h_a[\mathcal{C}_{\sigma_X}]}(a).$$

The amount of lost data in the interval $[0, t]$ is

$$L(t) = a(t) - x(t) = \sup_{k \geq 0} \left\{ \sup_{0 \leq s_{2k} \leq \dots \leq s_2 \leq s_1 \leq t} \left\{ \sum_{i=1}^k [a(s_{2i-1}) - a(s_{2i}) - \sigma(s_{2i-1} - s_{2i})] \right\} - kX \right\}.$$

As an application of this representation of $L(t)$, we show how to retrieve the bound on the loss rate obtained by Chuang and Chang [7], when we have an arrival curve α for the input flow a . Define

$$\hat{l}(t) = 1 - \inf_{0 < s \leq t} \frac{\sigma(s) + X}{\alpha(s)}. \quad (7.25)$$

Then for any $0 \leq u < v \leq t$, since $a(v) - a(u) \leq \alpha(v - u)$ by definition of an arrival curve,

$$1 - \hat{l}(t) = \inf_{0 < s \leq t} \frac{\sigma(s) + X}{\alpha(s)} \leq \frac{\sigma(v - u) + X}{\alpha(v - u)} \leq \frac{\sigma(v - u) + X}{a(v) - a(u)}.$$

Therefore, for any $0 \leq u \leq v \leq t$,

$$a(v) - a(u) - \sigma(v - u) - X \leq \hat{l}(t) \cdot [a(v) - a(u)].$$

For any integer $n \geq 1$, and any sequence $\{s_k\}_{1 \leq k \leq 2n}$, with $0 \leq s_{2n} \leq \dots \leq s_1 \leq t$, setting $v = s_{2i-1}$ and $u = s_{2i}$ in the previous equation, and summing over i , we obtain

$$\sum_{i=1}^n [a(s_{2i-1}) - a(s_{2i}) - \sigma(s_{2i-1} - s_{2i}) - X] \leq \hat{l}(t) \cdot \sum_{i=1}^n [a(s_{2i-1}) - a(s_{2i})],$$

from which we deduce that

$$L(t) \leq \sup_{n \in \mathbb{N}} \left\{ \sup_{0 \leq s_{2n} \leq \dots \leq s_1 \leq t} \left\{ \sum_{i=1}^n [a(s_{2i-1}) - a(s_{2i}) - \sigma(s_{2i-1} - s_{2i}) - X] \right\} \right\} \leq \hat{l}(t) \cdot a(t),$$

which shows that $\hat{l}(t) \geq l(t) = L(t)/a(t)$. To have a bound independent of time t , we take the sup over all t of (7.25):

$$\hat{l} = \sup_{t \geq 0} \hat{l}(t) = 1 - \inf_{t > 0} \frac{\sigma(t) + X}{\alpha(t)}.$$

Finally, the maximal solution of (6.22) in **Example 5** is, after similar manipulations as those made in Example 4,

$$x = \overline{\mathcal{P}_L \wedge \mathcal{C}_\sigma}(a) = \overline{\mathcal{P}_L[\mathcal{C}_\sigma]}(a).$$

One checks that x is L-packetized, and is thus the output of the packetized greedy shaper.

8 Conclusion

Network calculus belongs to what is sometimes called topical (or exotic) algebras, a set of mathematical results, often with high description complexity, but offering deep insights into man-made systems such as communication networks. This paper has underlined the importance of a simple fixed point residuation theorem in the networking context.

References

- [1] R. Agrawal, R. L. Cruz, C. Okino and R. Rajan, ‘Performance Bounds for Flow Control Protocols’, *IEEE Trans. on Networking*, vol 7(3), pp 310–323, June 1999.
- [2] R. Aggrawal and R. Rajan. ‘Performance bounds for guaranteed and adaptive services’, Technical report RC 20649, IBM, December 1996.
- [3] F. Baccelli, G. Cohen, G. J. Olsder and J.-P. Quadrat. *Synchronization and Linearity, An Algebra for Discrete Event Systems*, John Wiley and Sons, August 1992.
- [4] C.S. Chang. ‘A filtering theory for deterministic traffic regulation’, in *Proceedings Infocom’97*, Kobe, Japan, April 1997.
- [5] C.S. Chang, *Performance Guarantees in Communication Networks*, Springer-Verlag, New York, 2000.
- [6] C.S. Chang, R. L. Cruz, J. Y. Le Boudec, P. Thiran ‘A Min-Plus System Theory for Constrained Traffic Regulation and Dynamic Service Guarantees’, *IEEE/ACM Transactions on Networking*, 2002 (to appear).
- [7] C.-M. Chuang and J.-F. Chang, ‘Deterministic Loss Ratio Quality of Service Guarantees for High Speed Networks’, *IEEE Com. Letters*, vol. 4, pp. 236–238, July 2000.
- [8] R. L. Cruz, ‘Quality of service guarantees in virtual circuit switched networks’, *IEEE Journal on Selected Areas in Communication*, pp. 1048–1056, August 1995.
- [9] J. Y. Le Boudec, ‘Some properties of variable length packet shapers’ *Proceedings of ACM Sigmetrics 2001*, Boston, June 2001.
- [10] J. Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Springer-Verlag, vol. LCNS 2050, New York, 2001.
- [11] J. Y. Le Boudec and O. Verscheure, ‘Optimal Smoothing for Guaranteed Service’ *IEEE/ACM Transactions on Networking*, Vol. 8, pp. 689–696, Dec 2000.
- [12] P. Thiran, J.-Y. Le Boudec and F. Worm, ‘Network calculus applied to optimal multimedia smoothing’, *Proc of Infocom 2001*, April 2001.